i-PERCEPTION

a Pion publication

i-Perception (2014) volume 5, pages 75 –78

dx.doi.org/10.1068.i004ir ISSN 2041-6695 perceptionweb.com/i-perception

i-Review

Sharing code

Jonas Kubilius

Laboratories of Biological and Experimental Psychology, KU Leuven, Tiensestraat 102 bus 3714, B-3000 Leuven, Belgium; e-mail: jonas.kubilius@ppw.kuleuven.be

Received 3 February 2014; published 5 February 2014

Abstract. Sharing code is becoming increasingly important in the wake of Open Science. In this review I describe and compare two popular code-sharing utilities, GitHub and Open Science Framework (OSF). GitHub is a mature, industry-standard tool but lacks focus towards researchers. In comparison, OSF offers a one-stop solution for researchers but a lot of functionality is still under development. I conclude by listing alternative lesser-known tools for code and materials sharing.

Keywords: code sharing, github, open science framework, open science, version control.

1 Introduction

With the increase of interest in open access, psychologists are slowly coming to embrace the concept of "Open Science" whereby not only publications but also other research materials would be freely shared, including datasets, code of experiments and analyses, peer review and postpublication comments, and research ideas. In this review I compare two popular solutions for code sharing, namely, GitHub (https://github.com/) and the Open Science Framework (OSF) (https://opensciencefframework.org/; Nosek & Bar-Anan, 2012; Nosek et al., 2012), in the context of the needs of psychological community. While these two tools prove sufficiently versatile to accommodate most researchers needs, I list some further alternatives that might be useful to some scientists.

Table 1. Comparison of features that Open Science Framework (OSF) and GitHub offer.

Feature	OSF	GitHub
Private repositories	yes, including their components	paid option only
Space	?	unlimited
Version control	yes	yes
Study registration	yes	yes
Remote syncing / seamless backup	partial*	automatic
Attribution via forking	yes	yes
Usage statistics	yes	yes
Issue reports / comments	partial*	yes
License	?	any
Science oriented	yes	no
Digital object indentifier	planned	no
Nontext / large files	yes	not encouraged

* via GitHub add-on.

2 GitHub

WGitHub (http://github.com; see https://github.com/qbilius/psychopy_ext for an example repository) is a web service for hosting projects that rely on the git version control system (http://git-scm.com/). Originally intended for software developers, this service has been embraced by many users, including the academic community, as a simple yet powerful platform for sharing code.

In essence, GitHub offers a file hosting service in the cloud. Users can upload files, and other users can download them, or, depending on permissions, even modify them. Conveniently, GitHub offers a powerful web interface in order to facilitate file editing and collaboration. For example, an online editor allows people to contribute even if they do not have git or a GitHub client installed. It is also straightforward to report bugs or comment in the online interface. Contributors can explore usage statistics, such as a number views and unique visitors, contributions over time, or how many people are watching this repository.

However, GitHub offers much more than a mere file hosting service. In particular, its power lies in the implementation of the git distributed version (or revision) control. Initially, a user sets up a git repository where all files will reside using the git software. As the project keep evolving, a user commits his or her updates with a message explaining the changes. Thus, a timeline of project development is created, providing the possibility to undo changes at any point. The benefit is at least twofold. On the one hand, everything is continuously backed up and there is no fear anymore that collaborators might change or delete something essential. On the other hand, file organization is massively improved as users do not have to store multiple copies of the same file, as is often the case with manuscript revisions, for example.

Moreover, imagine that two users edit the same file at the same time and try to upload it to the online repository. In conventional file hosting or syncing solutions like Dropbox, either the most recent file ends up overwriting another user's copy, or two files appear. Such undesirable behavior is not possible in the version control setting. Each user has their own copy of the project which are not automatically synced across computers as in Dropbox. When a user pushes his or her changes to the online GitHub repository, git first inspects if any of the changes are in conflict with the existing files on GitHub, for example, due to the scenario described above. If a conflict is found, the user is prompted to resolve it by comparing the two files and manually selecting the relevant bits.

While such functionality is clearly handy and very powerful, it may easily become overwhelming for many beginners. To help get started, GitHub provides a very polished git client for Windows which should provide a smooth entry experience even for the least technologically savvy users. Most daily routines—such as creating repositories, adding and committing changes, and synchronizing with the remote repository—are available with just a click of a button in an extremely intuitive interface.

However, while an excellent choice for a day-to-day development of code, GitHub might not be an ideal choice for sharing research materials. In particular, by its nature, GitHub is meant to share the ongoing development of the code rather than some "final" version that many researchers tend to prefer, such as a manuscript and all associated materials at the point of publishing. While tagging a particular version is possible, intuitively GitHub still appears more malleable than a mere upload of a zip file somewhere.

Moreover, since it is not primarily designed for the academic community, GitHub does not offer many useful features specific for the needs of researchers. For example, organizing a study into individually shareable components is not possible, and a study registration is only partially available if users realize that it is similar to committing and tagging. Also, digital object identifiers (dois) or equivalent stable links are not supported.

To sum up, GitHub provides a robust platform for code sharing which should be simple enough to use even for inexperienced users. However, as GitHub was not really designed for academia, the OSF emerged as a more versatile and flexible platform for researchers.

3 The Open Science Framework

The OSF debuted in 2012 with an aim to increase sharing, collaboration, and transparency in research. In a nutshell, OSF provides a rapidly growing free online platform and tools for researchers to share their data and code and collaborate on projects. Notably, the OSF has been employed in the Reproducibility Project, an effort of more than 150 researchers to replicate a large number of publications in psychology (Open Science Collaboration, 2013; see https://osf.io/ezcuj/).

On the surface, GitHub users will recognize many similarities in OSF's logic; in fact, in the backend, both GitHub and OSF use the git version control system. A researcher can upload all research materials (for example, data, protocol, experimental scripts, analysis scripts, or the final manuscript) to the OSF website, and a description of project can be provided next to the uploaded files. (However, notice that, unlike in GitHub, uploading and updating files can only be done manually in OSF.) The project can also be subdivided into multiple components, each covering a particular step of research workflow, such as description of hypothesis, experimental and analysis code, and data. All changes to the project are recorded, providing the same version control functionality as in GitHub.

OSF is also aware of the common practice in academia to refer to some stable archival copy of the project with the exact materials at that point—for example, just after data collection or when a manuscript is submitted. To implement this, OSF provides an option to freeze the project, called "registration", which creates a read-only snapshot to the current state of the project with a unique hyperlink. Moreover, this registration idea proves particularly useful for keeping researchers honest about their planned analyses as they are encouraged to preregister their studies.

By design, the OSF also encourages collaboration on projects. Coworkers can be quickly added to the project with the click of a button, and the project itself can be chosen to be private or public. If it is public, then all materials, project's activity, and visitor statistics are visible on the project's home page. Moreover, a public project can be forked by another researcher to build upon it, and a credit will be assigned not only to this researcher but also to the original repository creator. In this manner, knowledge accumulation is inherently coupled with attribution and contribution traceability.

In fact, OSF might be better conceptualized as a hub of services for an academic user. Rather than offering a mere file repository with a few bells and whistles on top of GitHub, OSF aims at integrating various services under its hood so that information could be easily processed, combined, and exchanged. As a first example of the power of this approach, OSF introduced a GitHub add-on which embeds files stored in a chosen GitHub repository into your OSF project. Thus, users can benefit from all advantages that GitHub offers and yet operate within the same OSF project which is arguably better suited for sharing academic materials.

Interestingly, OSF also provides an incentive structure aimed at encouraging researchers to actively participate in sharing and collaboration. A user is rewarded with points for performing various actions on the website, and the total activity score is publically visible on the user's profile. Moreover, visitor statistics are automatically displayed, together with a number of forks and other people "watching" the project.

Currently, the major limitation of OSF is its immaturity (it is in beta). Its application programming interface is still under a heavy development, and there are few add-ons offered at the moment. Thus, for example, those researchers who want to take an advantage of OSF's version control cannot do it directly because data cannot be pushed or pulled automatically from the OSF's repositories.

Similarly, while it has been designed to improve sharing, at the moment it seemingly requires too much user effort and thus is effectively limited to Open Science enthusiasts. For example, materials can be uploaded only manually and downloaded only one by one. While for some researchers merely the possibility of sharing their materials might be sufficient, in collaborative environments this is an annoying limitation, and instead services such as Dropbox (http://dropbox.com) appear to be more intuitive and effective.

Finally, OSF's design currently lacks clarity and polish, making it less intuitive to get started. For example, because the main page of a project and its components are visually very similar, it becomes difficult to develop an overview of the project structure and not get lost while navigating.

Despite these limitations, OSF provides a research-oriented platform with an enormous potential in the future. Since users can already link their GitHub repositories with OSF, I would encourage setting up a OSF project, registering your studies, and further operating within GitHub.

4 Other tools

Many researchers will want to keep their code private until publication. While OSF is very much aware of this requirement, GitHub does not offer free plans for private repositories. However, a free solution for an unlimited number of private repositories is available from Bitbucket (http://bitbucket.org), another major code hosting website. Moreover, Bitbucket allows an unlimited number of collaborators for academic community. In fact, as compared with GitHub, Bitbucket falls short mostly in terms of its desktop client which appears to have a steeper learning curve, though GitHub's client can be used with Bitbucket repositories, effectively addressing this limitation. Another (albeit temporary) limitation is the lack of a Bitbucket add-on in OSF.

Other tools similar to GitHub in their functionality but more science-oriented include Banyan (http://banyan.co) and SciGit (http://scigit.com). In particular, SciGit aims to improve collaborating on manuscripts using the git version control system. For sharing the entire research workflow (i.e., not only code), figshare (http://figshare.com; full disclosure: I am a figshare advisor), and Zenodo (http://zenodo.org) are recommended.

Acknowledgments. Jonas Kubilius is a research assistant of the Research Foundation – Flanders (FWO).

References

Nosek, B. A., & Bar-Anan, Y. (2012). Scientific utopia: I. Opening scientific communication. *Psychological Inquiry*, *23*(3), 217–243. doi:10.1080/1047840X.2012.692215

Nosek, B. A., Spies, J. R., & Motyl, M. (2012). Scientific utopia II. Restructuring incentives and practices to promote truth over publishability. *Perspectives on Psychological Science*, *7*(6), 615–631. doi:10.1177/1745691612459058

Open Science Collaboration (2013). *The Reproducibility Project: a model of large-scale collaboration for empirical research on reproducibility* (3 January). doi:10.2139/ssrn.2195999